Investigating computational geometry for failure prognostics

Emmanuel Ramasso

FEMTO-ST Institute
UMR CNRS 6174 - UFC / ENSMM / UTBM, 25000, Besançon, France
emmanuel.ramasso@femto-st.fr

ABSTRACT

Prognostics and Health Management (PHM) is a multidisciplinary field aiming at maintaining physical systems in their optimal functioning conditions. The system under study is assumed to be monitored by sensors from which are obtained measurements reflecting the system's health state. A health indicator (HI) is estimated to feed a data-driven PHM solution developed to predict the remaining useful life (RUL). In this paper, the values taken by an HI are assumed imprecise (IHI). An IHI is interpreted as a planar figure called polygon and a case-based reasoning (CBR) approach is adapted to estimate the RUL. This adaptation makes use of computational geometry tools in order to estimate the nearest cases to a given testing instance. The proposed algorithm called RULCLIPPER is assessed and compared on datasets generated by the NASA's turbofan simulator (C-MAPSS) including the four turbofan testing datasets and the two testing datasets of the PHM'08 data challenge. These datasets represent 1360 testing instances and cover different realistic and difficult cases considering operating conditions and fault modes with unknown characteristics. The problem of feature selection, health indicator estimation, RUL fusion and ensembles are also tackled. The proposed algorithm is shown to be efficient with few parameter tuning on all datasets.

1. INTRODUCTION

Prognostics and Health Management (PHM) is a recent field of research perceived as a key process (Vachtsevanos, 2006) to increase the availability of equipments while decreasing maintenance costs. Many applications of PHM can be found, in particular for locomotives (Bonissone, Varma, & Aggour, 2005), fleet of vehicles (Saxena, Wu, & Vachtsevanos, 2005), bearings (He & Bechhoefer, 2008; Gelman, Patel, Murray, & Thomson, 2013), batteries (Saha, Goebel, Poll, & Christophersen, 2009; Orchard, Cerda, Olivares, & Silva, 2012), turbofan engine (T. Wang, 2010; T. Wang, Yu, Siegel, & Lee,

2008; Coble & Hines, 2011; P. Wang, Youn, & Hu, 2012), actuators (Goharrizi & Sepehri, 2010; Daigle & Goebel, 2011), wind turbines (Lapira et al., 2011; He, Bechhoefer, & Saxena, 2013), electro-mechanical systems (Gucik-Derigny, Outbib, & Ouladsine, 2011), fuel cells (Zhang & Pisu, 2014), electronics (Celaya, Kulkarni, Biswas, & Goebel, 2012), and structures (Sankararaman, Ling, Shantz, & Mahadevan, 2009; Mulligan, Yang, Quaegebeur, & Masson, 2013).

A PHM solution is called data-driven when the underlying models are built using sensor measurements while it is called physics-based when physical laws (thermodynamics, physics, mechanics and so on) are exploited to create the models. In this paper, a data-driven approach is proposed. Nevertheless, in both cases, it is crucial to represent the uncertainty and imprecision appropriately according to the underlying empirical information which is available (Beer, Ferson, & Kreinovich, 2013). For that, various techniques are available (Vachtsevanos, 2006), in particular the theory of probability (including Bayesian approaches, interval probabilities and random sets) (Saha, Goebel, & Christophersen, 2008; Orchard, Kacprzynski, Goebel, Saha, & Vachtsevanos, 2008), the Dempster-Shafer's theory of belief functions (Serir, Ramasso, & Zerhouni, 2013; Ramasso & Denoeux, 2013; Ramasso, Rombaut, & Zerhouni, 2013) and the set-membership approaches (including fuzzy sets and possibility theory) (Bonissone et al., 2005; Chen, Zhang, Vachtsevanos, & Orchard, 2011; Gouriveau & Zerhouni, 2012; Ramasso & Gouriveau, 2013). In PHM, the formulation of appropriate solutions should also take significant information into account but without introducing unwarranted assumptions to remain applicable and sufficiently general (Beer et al., 2013). The solutions should also cope with the quantity and quality of data that may have substantial impacts on results (Ramasso & Denoeux, 2013; Gouriveau, Ramasso, & Zerhouni, 2013).

Knowledge-based systems and Case-Based Reasoning approaches (CBR) have appeared as suitable tools for data-driven PHM (Saxena et al., 2005; T. Wang et al., 2008; T. Wang, 2010; Ramasso et al., 2013; Khelif, Malinowski, Morello, & Zerhouni, 2014; Ramasso, 2014). In CBR,

historical instances of the system - with condition data and known failure time - are used to create a library of degradation models or health indicators. Then, for a test instance, the similarity with the degradation models is evaluated generating a set of *Remaining Useful Life* (RUL) estimates which are finally aggregated. The required assumptions for CBR implementation are limited, the main issues consisting in, on the one hand, the choice of an appropriate similarity measure and, on the other hand, the selection of the relevant training instances. CBR approaches are also flexible since it is simple to incorporate quantitative and qualitative pieces of knowledge such as measurements and expertise.

We consider applications for which the noise due to various sources, such as operational conditions or fault modes, can not be well characterised and where filtering may change the meaning of the health indicator. We assume that the health indicator can not be well defined by a single real value but only by an *Imprecise Health Indicator* (IHI). The data points are supposed to represent vertices of a simple planar polygon: An IHI is thus a polygon-shaped signal represented by a planar figure. To fix ideas, an illustration taken from the turbofan engine dataset (Saxena, Goebel, Simon, & Eklund, 2008) (used and detailed in experiments) is given in Figure 1. The figure pictorially represents the IHI taken from the fourth dataset (made of two fault modes and six operating conditions) for the 8th training data (P_1), the 100th training data (P_2) and the 1st testing data (P_3) of this dataset. As depicted, fault modes may generate

- sudden changes in wear (e.g in P_1, $t \in [225, 275]$) that may increase the lifetime of the unit. It may be due to both fault modes and operating conditions, for example a drastic decrease of speed to cope with mechanical incidents or meteorological phenomenons.

- Unexpected changes in the trend, such as increasing instead of decreasing (e.g. P_2, $t > 125$) that may disturb the algorithm. It may be due to component failures such as sensors or electronics.

- Sudden bursts characterised by low signal-to-noise ratio (SNR) on a possibly short duration which deeply affect the HI (e.g. on P_3 with $t \in [10, 75]$) that may affect the lifetime accordingly to the fault type which is generally unknown.

Using computational geometry tools, a prognostics method is proposed to handle IHI without knowing nor estimating the noise properties. Performing prognostics in presence of IHI is tackled by using a CBR approach for which a similarity measure adapted to IHI is developed. The set of cases is made of training instances represented by polygons and the similarity with a testing instance recorded on the in-service system is made dependent on the degree of *intersection* between both training and testing polygon instances. The prognostics algorithm introduced is called "RULCLIPPER" (Remaining
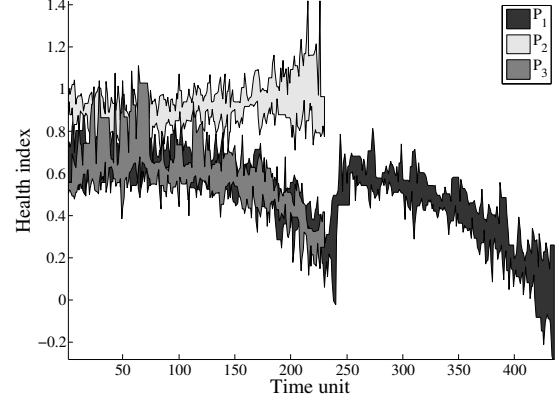

Figure 1. Effect of fault modes and operating conditions on health indicators estimation. HIs (here obtained from training instances) are described with planar figures called polygons.

Useful Life estimation based on impreCise heaLth Indicator modeled by Planar Polygons and similarity-basEd Reasoning") in reference to a widely used process in the Computer Graphics community called polygon clipping (Rosen, 2004).

The next Section is dedicated to the presentation of a methodology to build IHI and perform prognostics. The methodology is then applied on several datasets with fault modes and operating conditions and compared to other approaches (Section 3). An alphabetical list of terms used in the sequel is available in a glossary located at the end of this paper.

2. PROGNOSTICS BASED ON IHI AND CBR

A health indicator (HI) takes the form of a 1-dimensional real-valued signal $H = [x_1 \ x_2 \ldots \ x_j \ldots \ x_T]^{\mathrm{T}}, x_j \in \mathbb{R}$ obtained at some instants $t_1, \ t_2 \ldots \ t_T$.

2.1. Polygon-shaped representation of IHI

An IHI is defined as a polygon where each vertex is represented by a point (x_j, t_j) estimated from the original HI where x_j is the value of the HI at time t_j. The number of points is equal to $2 \cdot T$ and each of them belongs either to the *upper* or the *lower* envelope of the health indicator, where each envelope is made of exactly T points. In presence of high noise level, the extraction of the upper and lower envelopes is made easier by first filtering the original HI. The filtering also decreases the number of self-intersections of segments defined by consecutive vertices. The filter used in experiments paper was a 15-point moving average and may be stronger or softer according to the application considered.

Given the filtered health indicator denoted $\tilde{H} = [\tilde{x}_1 \ \tilde{x}_2 \ldots \ \tilde{x}_j \ldots \ \tilde{x}_T]^{\mathrm{T}}$, the upper envelope $\mathcal{S}$ is defined by:

$$\mathcal{S} = \{(x_j, t_j)|x_j \geq \tilde{x}_j\} \cup \{(x_{j-1}, t_j)|x_j < \tilde{x}_j\}, \quad (1)$$

meaning that, for a given data point j, if the HI value x_j at

2

time t_j is greater than the filtered value $\tilde{x}_j$ then the upper envelope is equal to the HI value, otherwise it takes its previous value. The lower envelope $\mathcal{I}$ is defined similarly by

$$\mathcal{I} = \{(x_j, t_j)|x_j < \tilde{x}_j\} \cup \{(x_{j-1}, t_j)|x_j \geq \tilde{x}_j\} . \quad (2)$$

Example 1 *An example of envelope computation is given in Figure 2 where the health indicator is decomposed into a lower envelope (circle) and upper envelope (stars) around the smooth HI (solid line).*
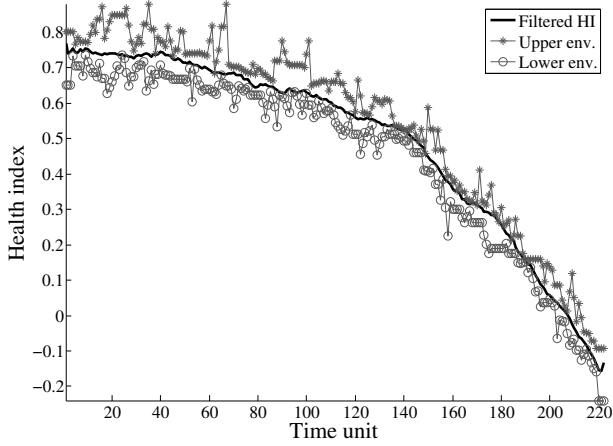

Figure 2. Illustration of envelope computation.

The ordered pairs of vertices listed counterclockwise represents a bounding closed polygonal chain that separates the plane into two regions. The word "polygon" refers to a plane figure bounded by the closed path defined as:

$$\mathcal{P} = \{(x_1, t_1)^{\mathcal{S}}, (x_2, t_2)^{\mathcal{S}} \ldots (x_j, t_j)^{\mathcal{S}} \ldots (x_T, t_T)^{\mathcal{S}},$$
$$(x_T, t_T)^{\mathcal{I}}, (x_{T-1}, t_{T-1})^{\mathcal{I}} \ldots (x_1, t_1)^{\mathcal{I}}, (x_1, t_1)^{\mathcal{S}}\} \quad (3)$$

with $(x_j, t_j)^{\mathcal{S}} \in \mathcal{S}$ and $(x_j, t_j)^{\mathcal{I}} \in \mathcal{I}$. To close the polygon, the first and last vertices are the same. The pairs of vertices define a finite sequence of straight line segments representing the polygon.

More specifically, a polygon is a region of the plane enclosed by a simple cycle of straight line segments where nonadjacent segments do not intersect and two adjacent segments intersect only at their common endpoint (Rosen, 2004). However, the second part of the definition of the bounds may generate some segment intersections. These inconsistencies can be corrected easily by exchanging the corresponding values of the lower and upper bounds when an intersection is detected. When consistent bounds are obtained, the polygon is made of non-intersecting line segments which characterise a Jordan's simple closed curve also called *simple* polygon (Filippov, 1950). This category of polygon enables one to apply some standard algorithms from Computational Geometry (Rigaux, Scholl, & Voisard, 2002; Rosen, 2004; Longley, de

Smith, & Goodchild, 2007). Note that some of the most efficient algorithms for operations on polygons can manage self-intersections (Vatti, 1992; Greiner & Hormann, 1998) but these inconsistencies generally increase time-consumption.

2.2. CBR approach for prognostics based on IHI

2.2.1. Training dataset

We assume the training dataset to be composed of N training instances:

$$\mathcal{L} = \{\mathcal{P}_i, \mathcal{K}_i\}_{i=1}^N \quad (4)$$

where $\mathcal{P}_i$ is the ith polygon attached to the ith imprecise health indicator H_i and $\mathcal{K}_i = [y_1 \ y_2 \ldots y_j \ldots y_T]^{\mathrm{T}}, y_j \in \mathbb{N}$ represents a discrete-valued signal reflecting the system's health state. The component $\mathcal{K}_i$ may be useful in some applications where the system can be described by means of latent variables (Ramasso & Denoeux, 2013; Javed, Gouriveau, & Zerhouni, 2013). In that case, $\mathcal{K}_i$ may represent a partial knowledge about the state. For example, in (Ramasso et al., 2013), partial knowledge was encoded by belief functions to express imprecision and uncertainty about the states.

Example 2 *An illustration of the use of discrete and continuous information (Ramasso et al., 2013) is depicted in Figure 3 for the same health indicator as in Figure 2. The states $\mathcal{K}_i$ for the ith health indicator H_i represent degradation levels and can be automatically found by applying a clustering algorithm (here the Gustafson-Kessel (Gustafson & Kessel, 1978)) taking as input the HI shown in Figure 2 (solid line) and run with 10 states. The transition to the last state means that the end-of-life is approaching.*
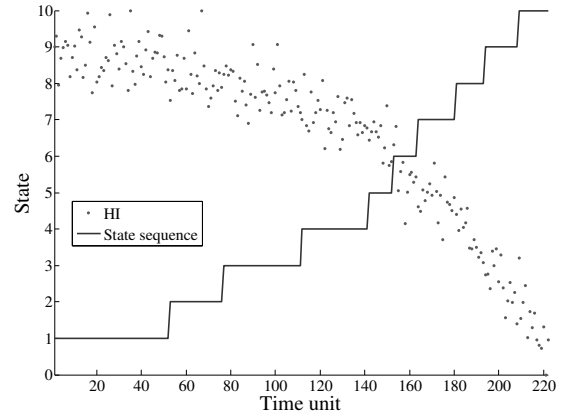

Figure 3. Segmentation into degradation levels (same HI as in Fig. 2).

2.2.2. Determining the nearest case

A testing instance takes the form of a health indicator H^* from which the envelopes can be estimated as explained in the previous paragraph, leading to the polygon representation $\mathcal{P}_*$.

3

As in usual CBR approaches for prognostics (T. Wang, 2010), the goal is to sort the training instances with respect to their similarity to the testing instance. However, since the training instances are made of polygons, usual distance measures are not adapted.

Getting inspired from the Computer Vision community (Powers, 2011), *recall*, *precision* and F_β indices are used to quantify the relevance of a training instance compared to the testing one. Precision represents the fraction of the retrieved instance that is relevant, while recall is the fraction of the relevant instance that is retrieved. The F_β is an harmonic mean which gives equal weight to recall and precision when $\beta = 1$. Note that the three indices are normalised into $[0, 1]$.

More precisely, for the ith training instance:

1. Estimate the area of the intersection between the polygon $\mathcal{P}_i$ and $\mathcal{P}_*$:
$$\mathcal{A}_\cap = \text{Area}\,(\mathcal{P}_i \cap \mathcal{P}_*) \tag{5}$$

2. Compute the "recall":
$$R_{\text{ec}} = \frac{\mathcal{A}_\cap}{\mathcal{A}_i} \tag{6}$$

3. Compute the "precision":
$$P_{\text{rec}} = \frac{\mathcal{A}_\cap}{\mathcal{A}_*} \tag{7}$$

4. Compute the "$F_{\beta,i}$", in particular for $\beta = 1$, characterizing the similarity with the ith training instance:
$$F_{1,i} = 2 \frac{R_{\text{ec}} \cdot P_{\text{rec}}}{R_{\text{ec}} + P_{\text{rec}}} \tag{8}$$

where $\mathcal{A}_i, \mathcal{A}_*, \mathcal{A}_\cap$ represent the area of polygons $\mathcal{P}_i$, $\mathcal{P}_*$ and of their intersection respectively.

Example 3 *An illustration of intersection is given in Figure 4 where the darkest polygon represents a training instance and the two other polygons are testing instances. The whitest polygon is within the testing instance meaning that the precision is high, but the recall is pretty low since it covers only a small part of the testing instance. On the opposite, the third polygon covers entirely the testing instance leading to a high recall but its scattering decreases the precision.*

Practically, intersection construction is the main difficulty and was tackled quite recently in computational geometry for arbitrary planar polygons. It consists in determining the region of geometric intersection which can be performed in three phases (Rosen, 2004) (Chap. 38):

1. Compute the intersection between the boundaries of the objects using the linearithmic plane sweep algorithm (Bentley & Ottmann, 1979);

2. If the boundaries do not intersect then determine whether one object is nested within the other;
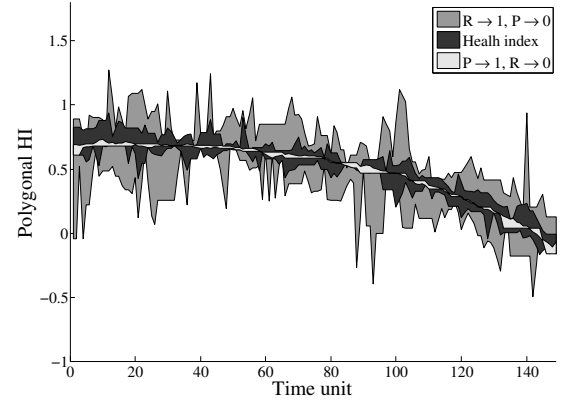

Figure 4. Illustration of recall and precision.

3. If the boundaries do intersect then classify the resulting boundary fragments gathered to create the final intersection region (Margalit & Knott, 1989; Chazelle & Edelsbrunner, 1992), which can be performed in linearithmic time. Regularized Boolean operations ensure the closure of the interior of the set-theoretic intersection.

In this paper, the Vatti's algorithm (Vatti, 1992) has been used because it is generic and can manage most of pratical cases. Several implementations of this algorithm have been proposed, especially in (Greiner & Hormann, 1998) which was shown to be particularly efficient. Several implementations can be found, in particular the GPC library available at `http://www.cs.man.ac.uk/~toby/gpc/` which proposes a flexible and highly robust polygon set operations library for use with *C*, *C#*, *Delphi*, *Java*, *Perl*, *Python*, and *Matlab* (version above 7-R14SP1).

2.2.3. Estimating the Remaining Useful Life (RUL)

The F_1 measure is used to sort the N training polygon instances in descending order: $\mathcal{P}_{(1)} > \mathcal{P}_{(2)} \cdots > \mathcal{P}_{(j)} \cdots > \mathcal{P}_{(N)}$ so that $\mathcal{P}_{(1)}$ is the closest instance to the testing instance and $\mathcal{P}_{(N)}$ the farthest one. The index (i) in $\mathcal{P}_{(i)}$ represents a reordering and the symbol ">" in "$\mathcal{P}_{(i)} > \mathcal{P}_{(j)}$" means that the ith polygon is more similar to the testing instance that the jth one.

CBR assumes that a limited number of instances, say K, are enough to approximate the testing instance. The K closest training instances can then be combined to estimate the RUL. The length of a training instance minus the length of a testing instance provides an estimation of the RUL (Figure 5). Given the definition of a polygon (Section 2.1) and of the training dataset (Eq. 4), the length of both the training and testing polygon instances is given by T_i and T_* respectively. Therefore, the estimated RUL is given by

$$\hat{\text{RUL}} = T_i - T_* \,. \tag{9}$$

4

Example 4 *Two polygons are illustrated in Figure 5, one coming from the training dataset #1 (the tenth instance) and one from the testing dataset #1 (the first instance). Since $T_1 = 222$ and $T_* = 31$, the estimated RUL is 191 time-units.*
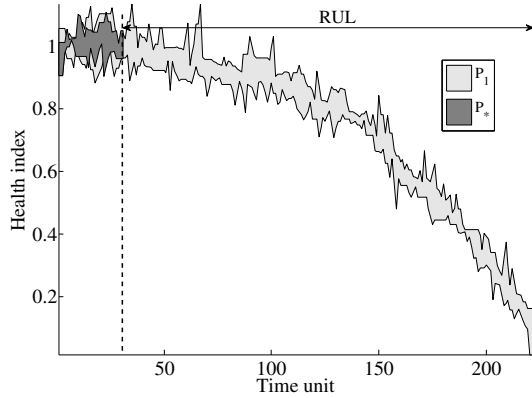

Figure 5. Polygon instances: training ($\mathcal{P}_1$) and testing ($\mathcal{P}_*$).

Each closest training instance $\mathcal{P}_{(i)}$ can be accompanied by a state sequence $\mathcal{K}_{(i)}$ so that K estimations of the RUL, denoted $\hat{\text{RUL}}_\mathcal{K}$, can be obtained from the state sequences in addition to the ones obtained with $\mathcal{P}_{(i)}$ and denoted $\hat{\text{RUL}}_\mathcal{P}$. Using $\mathcal{K}_{(i)}$, the last transition in the sequence is supposed to represent a jump of the system to a faulty state. This assumption relies on the fact that the last part of a training instance represents the system's end-of-life (Ramasso et al., 2013; Ramasso & Gouriveau, 2013; Javed et al., 2013).

The $2K$ estimations of the RUL can then be pooled in one set: $\hat{\text{RUL}}_{\mathcal{PK}} = \{\hat{\text{RUL}}_\mathcal{P}, \hat{\text{RUL}}_\mathcal{K}\}$ and an information fusion process can then be performed to combine these partial RUL estimates. According to the application, the fusion rule can be adapted (Kuncheva, 2004; T. Wang et al., 2008; Ramasso & Gouriveau, 2013). The fusion rules used in this paper will be presented in details in Section 3 (dedicated to the experiments).

A plot chart of the proposed methodology is depicted in Figure 6. The RULCLIPPER algorithm (in the dashed box) follows the steps presented in the previous sections. The remaining elements are common to other prognostics approaches based on health indicators, in particular the key paper presented in (T. Wang, 2010) where health indicators are defined conditionnally on operating conditions. These elements will be illustrated in the next section dedicated to experiments.

3. EXPERIMENTS: METHOD

RULCLIPPER is tested on the datasets obtained from the turbofan engine degradation simulator (Saxena, Goebel, et al., 2008). Before presenting results, several details about the datasets have to be presented, in particular how to select the

features and how to compute the health indicator.

3.1. Turbofan engine degradation simulator

The simulation model (Saxena, Goebel, et al., 2008) was built on the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) developed at NASA Army Research Lab., able to simulate the operation of an engine model of the 90.000 lb thrust class. A total of 21 output variables were recorded to simulate sensor measurements to the system. Another 3 variables representing the engine operating conditions were recorded, namely altitude (kilo feet), Mach number (speed) and Throttle Resolver Angle (TRA) value which is the angular deflection of the pilot's power lever having a range from 20% to 100%. In the sequel, references to variables are made by using their column position in the data files as provided on the data repository of the Prognostics Center of Excellence website: it begins by number 6 and finishes to 26 (see (Saxena, Goebel, et al., 2008) for details).

3.2. Datasets

Six datasets generated from independent simulation experiments were provided, each with some specificities (Saxena, Goebel, et al., 2008).

Datasets #1 and #2 include only one fault modes (HPC degradation) while datasets #3 and #4 include two (HPC degradation and fan degradation). Datasets #1 and #3 include a single operational condition against six for datasets #2 and #4. Dataset #4 represents the most complex case study. Datasets $#5_T$ (semi-final **t**esting dataset) and $#5_V$ (final **v**alidation dataset) were generated for the 2008's PHM data challenge with one fault mode and six operating conditions. It is important to emphasize that the two last datasets have *common training instances*. A summary of the six datasets are shown in Table 1 according to information taken from (Saxena, Goebel, et al., 2008).

Each dataset is divided into training and testing subsets. The training set includes instances with complete run-to-failure data (to develop life prediction models), and the actual failure mode for training instances in #3 and #4 is not labeled. The testing datasets include instances with data up to a certain cycle and are used for RUL estimation and algorithm performance evaluation.

The testing instances are also simulated run-to-failure and only an *earlier portion* of the history is provided. The actual life (RUL) of the testing instances are known only for datasets #1, #2, #3 and #4 and can be used for testing the algorithms. For datasets $#5_T$ and $#5_V$, results have to be uploaded to the data repository: uploading is allowed only once a day for $#5_T$ whereas only a single try is possible for dataset $#5_V$.

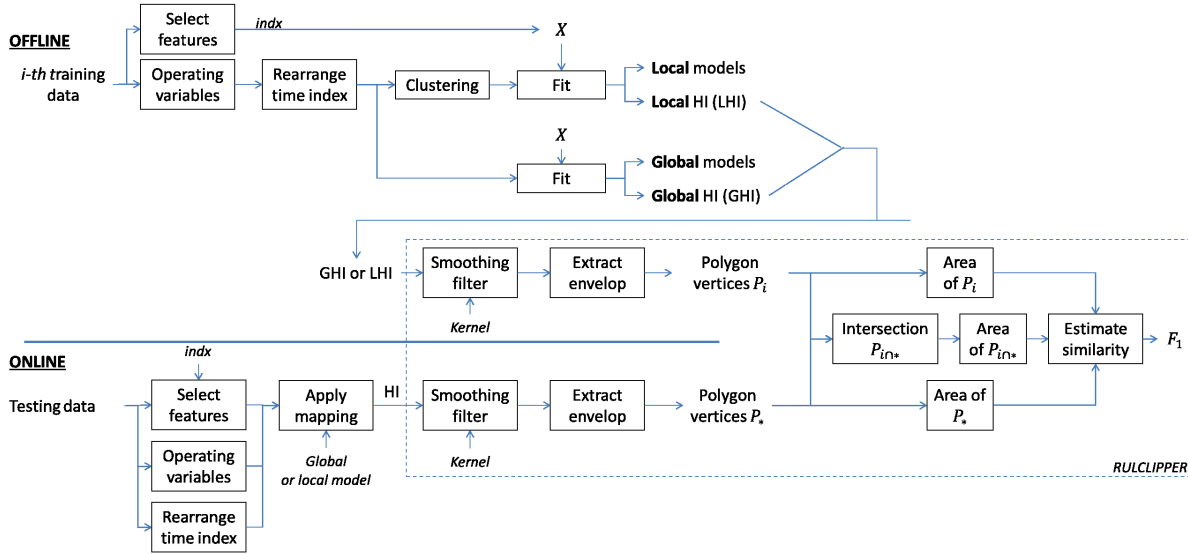The validation can be performed by many performance mea-

Figure 6. The sequence of operations involved in the proposed approach.

| Datasets | C-MAPSS DATASETS | | | | | |
| | TURBFOFAN | | | | CHALLENGE | |
	#1	#2	#3	#4	#5$_T$	#5$_V$
Nb. of faults	1	1	2	2	1	1
Nb. of operating conditions	1	6	1	6	6	6
Nb. training instances	100	260	100	249	218	
Nb. testing instances	100	259	100	248	218	435
Minimum RUL	7	6	6	6	10	6
Maximum RUL	145	194	145	195	150	190

Table 1. Datasets characteristics according to the organisers. In this paper, results for all datasets are provided. Note that the datasets called "data challenge" have a common training datasets made of 218 instances. The "semi-final" testing dataset (#5$_T$) is made of 218 instances and the "final" validation dataset (#5$_V$) is made of 435 instances.

sures (Saxena, Celaya, et al., 2008) among which accuracy-based measures such as the *timeliness*, also called scoring function in the sequel since it has been used in the data challenge to sort participants' algorithm. The review of papers using the C-MAPSS datasets show that the timeliness was the most used performance measure (about 30% of papers). Note that, for datasets #5$_T$ and #5$_V$, this performance measure is returned for each submission by the data challenge chairs.

For comparison purpose, the scoring function is also used in this paper with the same parameters as in the challenge:

$$S = \sum_{n=1}^{N} S_n \tag{10a}$$

$$S_n = \begin{cases} e^{-d_n/13} - 1, d_n \leq 0 \\ e^{d_n/10} \ - 1, d_n > 0 \end{cases}, n = 1 \dots N \tag{10b}$$

$$d_n = \text{estimated RUL} - \text{true RUL} \tag{10c}$$

This function, that assigns higher penalty to late predictions, has to be minimised. In addition to the scoring function (computed for all datasets), a second performance measure was

used (on datasets #1 to #4 for which we know the RUL) called accuracy measure A that evaluates the percentage of testing instances for which the RUL estimate falls within the interval $[-13, +10]$ around the true RUL (Saxena, Celaya, et al., 2008). These values are the same as the scoring function and was used in several papers such as (Ramasso et al., 2013) for dataset #1.

3.3. Related results on C-MAPSS

For comparison purpose, results of predictions from other researchers (as exhaustive as possible) are summarised below for each dataset. References have been put on the NASA PCOE website and a survey of papers in under preparation.

To our knowledge, the full testing dataset of #1 was only used in three papers: In (Liu, Gebraeel, & Shi, 2013) where the authors reported results by using an average error between true RUL and prediction; In the EVIPRO algorithm (Ramasso et al., 2013) and in (Khelif et al., 2014) where the performance was assessed by using the accuracy measure which was equal to 53% and 54% respectively on the testing dataset #1. The full testing datasets of #2, #3, #4 were not used

6

in the past (only a few instances were considered in a few papers).

Testing datasets $\#5_T$ (corresponding to a "semi-final" testing dataset) and $\#5_V$ (corresponding to the "final" validation dataset) represent datasets for which the true RULs is *not known*. These datasets were used in many papers summarised in Table 2 (for published work after 2008) and in Table 3 (for results of challengers during the competition in 2008). The complete review of scores on these datasets were found on the web or obtained by request to the conference chairs. In Table 3, methods (1), (2) and (3) were published in (T. Wang et al., 2008), (Heimes, 2008) and (Peel, 2008) respectively.

It can be observed that no score has been mentioned in the literature on the final validation dataset $\#5_V$ since 2008, whereas the semi-final testing dataset $\#5_T$ was used in several papers. The final dataset is complex and the performances obtained by the challengers are high. According to our knowledge, good performances (in terms of scoring) can be obtained on the final dataset only if the algorithm is robust. Indeed, a few important mistakes (too late or too early predictions) can lead to bad scores. This was also observed with RULCLIPPER on the other datasets. Robustness can be evaluated by computing several PHM metrics (Saxena, Celaya, et al., 2008) as proposed in (T. Wang, 2010).

Therefore, the generalisation capability of the algorithm should be ensured before trying the final dataset. This is illustrated in Tables 2-3 and Figure 15 which depict the scores obtained on the semi-final dataset $\#5_T$ and on the final dataset $\#5_V$. Some algorithms exhibited very low score on $\#5_T$ (made of 218 instances), whereas a relatively poor score was obtained on the final dataset. The winner obtained 737 on $\#5_T$ (according to the conference chairs) which is not the best score, but only 5636 on the final dataset $\#5_V$.

Algo. (pseudo.)	$\#5_T$	$\#5_V$
RULCLIPPER	**752**	**11672**
SBL (P. Wang et al., 2012)	1139	n.a.
DW (Hu, Youn, Wang, & Yoon, 2012)	1334	n.a.
OW (Hu et al., 2012)	1349	n.a.
MLP (Riad, Elminir, & Elattar, 2010)	1540	n.a.
AW (Hu et al., 2012)	1863	n.a.
SVM-SBI (Hu et al., 2012)	2047	n.a.
RVM-SBI (Hu et al., 2012)	2230	n.a.
EXP-SBI (Hu et al., 2012)	2282	n.a.
GPM3 (Coble, 2010)	2500	n.a.
RNN (Hu et al., 2012)	4390	n.a.
REG2 (Riad et al., 2010)	6877	n.a.
GPM2B (Coble, 2010)	19200	n.a.
GPM2 (Coble, 2010)	20600	n.a.
GPM1 (Coble, 2010)	22500	n.a.
QUAD (Hu et al., 2012)	53846	n.a.

Table 2. Performance of the state-of-the-art approaches on $\#5_T$ (semi-final dataset) and $\#5_V$ (final dataset) after 2008 (published work). See references and glossary for more details about the approaches.

Algo. (pseudo.) / Data	$\#5_T$	$\#5_V$
heracles (1)	737 (3rd)	5636 (1st)
FOH (2)	512 (2nd)	6691 (2nd)
LP (3)	n.a.	25921
sunbea	436.8 (1st)	54437 (22nd)
bobosir	1263	8637
L6	1051	9530
GoNavy	1075	10571
beck1903	1049	14275
Sentient	809	19148
A	975	20471
mjhutk	2430	30861
RelRes	1966	35863
phmnrc	2399	35953
SuperSiegel	1139	154999

Table 3. Pseudonyms and scores (known on *both* $\#5_T$ and $\#5_V$) during the 2008's PHM data challenge. Methods (1), (2) and (3) were published.

3.4. Priors about the datasets

Some rules were used to improve prognostics on these datasets, some have been proposed in previous papers:

R1: The first rule is related to the fact that, according to (Saxena, Goebel, et al., 2008), the maximum RUL in testing instances for $\#5_T$ was greater than 10 and lower than 150 time-units, while being greater than 6 and greater than 190 in testing instances for $\#5_V$. Moreover, most of previous approaches agreed on limiting the RUL estimates around 135 (depending on papers (T. Wang et al., 2008; T. Wang, 2010; Heimes, 2008; Riad et al., 2010)) because too large and late estimates are greatly penalized by the scoring function. So, for most of tests presented below, the RUL was given by $\max(6, \min(\hat{\text{RUL}}, 135))$.

R2: The difference between 1 and the average of the first 5% of an instance was used as an offset to compel the health indicator (HI) to begin around 1. Even though the health indicator function (Eq. 12) already compels it, there are some cases, in particular for $\#2$, $\#3$ and $\#4$, for which the health indicator was strongly disturbed by fault modes and operating conditions.

R3: To limit the impact of fault modes and to circumvent too early and late predictions, a detection of the monotonicity is performed. Monotonicity was pointed out as a key point in prognostics in particular in (Coble, 2010). The rule works as follows: When the testing instance is less than the half of the training instance, then there is a risk of early or late predictions. In that case, starting from the end of the testing instance, if more than 25 consecutive samples are above (resp. under) the training instance then it is assumed that this instance will be responsible of too early (resp. too late) predictions. In that case, the training instance is not taken into account for RUL estimation. This simple rule was applied as such on all datasets considered (even without fault modes or

without operating conditions).

R4: To decrease the risk of overpredictions, the sequence of states $\mathcal{K}$ were made of two states, the second state being activated only 15 samples before the end-of-life. This setting similar to (Ramasso & Gouriveau, 2013) and was the same for all tests and all datasets.

These rules have been developed specifically for the C-MAPSS datasets and can probably not generalise to other ones. The RULCLIPPER algorithm remains general enough to be applied on other datasets with other specific rules.

3.5. Local/global health indicator estimation

To reflect a real-world and practical cases, the health indicators (HI) for both training and testing datasets were not given by the organisers (Saxena, Goebel, et al., 2008). An adaptation of the approach proposed in (T. Wang, 2010) is presented below to estimate the HI for each instance. These HIs (highly corrupted by noise) are the basis of the proposed methodology described in previous sections (Fig. 6).

The set of features for the ith unit is $\mathbf{X}_i = [\mathbf{x}_1 \ \mathbf{x}_2 \ldots \mathbf{x}_t \ldots \mathbf{x}_{T_i}]^{\mathsf{T}}$ where $\mathbf{x}_t = [x_{t,1} \ x_{t,2} \ldots x_{t,m} \ldots x_{t,q}]$ is the q-dimensional feature vector at t (composed of sensor measurements), and $\mathbf{u}_t$ is the vector of operating conditions at t. The latter can be clustered into a finite number of operating regimes (T. Wang, 2010; Richter, 2012). Crisp outputs are obtained such that the current regime at time t, C_t, is precisely known. Then, for samples $(\mathbf{u}_t, \mathbf{x}_t)$ collected at early age of the system, e.g. $t < \sigma_1$, the health indicator attached to the ith training unit is $\mathrm{HI}(\mathbf{x}_t, \boldsymbol{\theta}^p) = 1$, where the set of parameters $\boldsymbol{\theta}^p$ depends on the model used to link regimes and sensor measurements.

At late age of the system, e.g. $t > \sigma_2$, the corresponding output is $\mathrm{HI}(\mathbf{x}_t, \boldsymbol{\theta}^p) = 0$. In (T. Wang, 2010), the author used only the data at $t > \sigma_2$ and $t < \sigma_1$ in addition to 6 models (one for each operating mode) built on all data. In comparison, we propose to make use of samples between σ_1 and σ_2 while building a local model for each operating mode in each training instance. Moreover, we have used one HI for each training instance while in (T. Wang, 2010) a global HI model was estimated using all instances.

The corresponding output of the HI is set to (Figure 7):

$$\hat{\mathrm{HI}}_i(\mathbf{x}_t, \boldsymbol{\theta}^p) \equiv 1 - \exp\left(\frac{\log(0.05)}{0.95 \cdot T_i} \cdot t\right), t \in [\sigma_1, \sigma_2]. \quad (11)$$

This function allows to compel the health indicator to be globally decreasing, from 1 (healthy) to 0 (faulty). As proposed in (T. Wang, 2010), $\sigma_1 = T_i \cdot 5\%$ and $\sigma_2 = T_i \cdot 95\%$ where T_i is the length of the ith training instance. We used local linear models for multi-regime health assessment so that $\boldsymbol{\theta}_i^p = [\theta_{i,0}^p \ \theta_{i,1}^p \ldots \theta_{i,q}^p]$ represents the parameters of a linear model defined conditionnally to the pth regime (Figure 8).
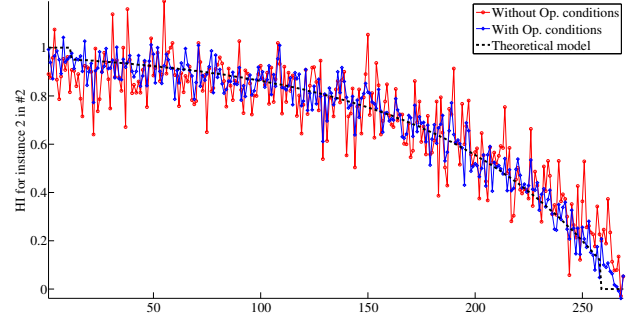

Figure 7. The theoretical model (Eq. 11) of degradation: Surimposed to HI estimated with and without operating conditions illustrating their impact on the HI
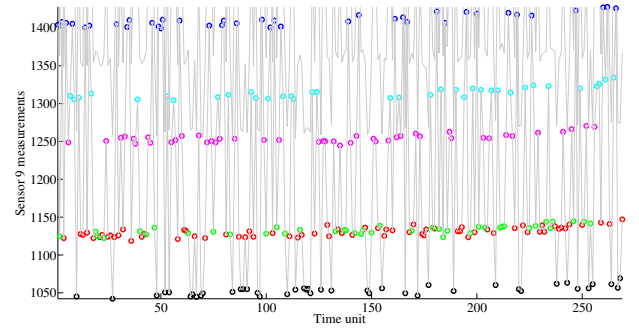

Figure 8. Operating conditions in each regime: Sensor measurements are locally linear as assumed in Eq. 12

The health indicator at time t given the pth regime can thus be estimated as

$$\mathrm{HI}_i(\mathbf{x}_t, \boldsymbol{\theta}_i^p) = \theta_{i,0}^p + \sum_{n=1}^{q} \theta_{i,n}^p \cdot x_{t,n} \quad (12)$$

where $\boldsymbol{\theta}^p$ can be estimated by standard least-squares algorithms. In experiments, in case the estimation of HI is performed by considering the three operating conditions, then it will be called a *local* approach (Fig. 6) and *global* otherwise. HIs are then transformed into IHIs as proposed in previous sections (Fig. 6). An example of HI estimation using local and global approaches are illustrated in Figure 9.

3.6. Information fusion for improved RUL estimation

The first family of rules is a combination of minimum and maximum RUL estimates suggested in (T. Wang, 2010):

$$\alpha m M(R) = \alpha \cdot \min R + (1 - \alpha) \cdot \max R \quad (13)$$

where R is a set of RUL estimates and $\alpha m M(R)$ the combination result. For example, in (T. Wang, 2010), $\alpha = 13/23$. In this paper, we considered $\alpha \in \{0.1, 0.2, 0.3, \ldots 0.9, 13/23\}$. The authors in (T. Wang, 2010) also added two outlier removal (OR) rules to keep
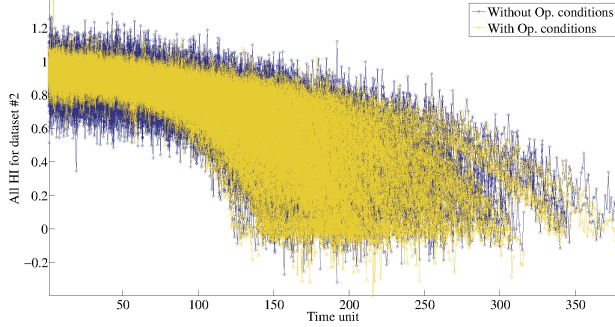
Figure 9. All HI estimated for training dataset #2 with and without operating conditions: High scattering in terms of initial wear, degree of degradation and RUL. The scattering is strongly reduced when taking operating conditions into account.

RULs within the interquartile range:

$$
\begin{aligned}
OR \quad &: \quad \{a \in R : a \in [q_{25}, q_{75}]\} \\
WL \quad &: \quad \{a \in R : inf < a < sup\} \\
&\quad inf = q_{50} - 3 \cdot (q_{50} - q_{25}) \\
&\quad sup = q_{50} + 2 \cdot (q_{75} - q_{50})
\end{aligned}
\tag{14}
$$

The set of RUL estimates considering either discrete ($\mathcal{K}$) or continuous predictions ($\mathcal{P}$), is denoted

$$
R \equiv \hat{\mathrm{RUL}}_{\mathcal{P}[\mathcal{K}]}^{[OR|WL],[th],M}
\tag{15}
$$

Only the M first RULs estimates were taken into account (sorted according to the F_1 measure) with $M \in \{11, 15\}$ in this study. $OR|WL$ means that one of the outlier removal operators was applied. The optional parameter $[th]$ means that only training instances with F_1 measure greater than 0.5 were kept.

Weighted average is the second family of rules:

$$
mw_L^{[e],[OR]} = \sum_{i=1}^{L} \omega_i^{[e],[OR]} \cdot R_{(i)}
\tag{16}
$$

where the weights are made dependent on the similarity $F_{1,i}$ (Eq. 8) between the testing instance and the ith training instance; $R_{(i)}$ is the ith RUL estimate in set of RULs R sorted in descending order with respect to the similarity ($F_{1,i}$) ; $L \in \{3, 5, 7, 9, 11, 15\}$ is the number of RULs kept to compute the average while applying or not the outlier removal rule OR. The weights are given by the following equations:

$$
\omega_i = F_{1,i} / \sum_{k=1}^{L} F_{1,k} ,
\tag{17}
$$

with softmax normalisation:

$$
\omega_i^e = \exp(F_{1,i}) / \sum_{k=1}^{L} \exp(F_{1,k}) ,
\tag{18}
$$

using outlier removal (OR):

$$
\omega_i^{OR} = OR(F_{1,i}) / \sum_{k=1}^{L} OR(F_{1,k}) ,
\tag{19}
$$

and combining OR and softmax:

$$
\omega_i^{e,OR} = \exp(OR(F_{1,i})) / \sum_{k=1}^{L} \exp(OR(F_{1,k})) .
\tag{20}
$$

The third kind of rules is a combination of the previous ones:

$$
\hat{\mathrm{RUL}} = 0.5 \cdot \alpha m M(R) + 0.5 \cdot m w_L^{[e],[OR|WL]}
\tag{21}
$$

Considering several combinations of parameters, about 3168 rules were considered.

3.7. Selecting the subset of sensors

As shown by the literature review presented beforehand, many combinations of features can be used (among 21 variables), and a subset was particularly used made of features $\{7, 8, 12, 16, 17, 20\}$ (involving key sensors for the turbofan degradation (Sarkar, Jin, & Ray, 2011)). To this preselection, a subset of sensors was added from every possible subsets with cardinality equal to 1, 2, 3 and 4 in $\{\emptyset, 9, 10, 11, 13, 14, 18, 19, 22, 25, 26\}$ as well as subsets of cardinality 5 comprising sensor 9 leading to a total of 511 cases. For each combination (511 cases for each dataset), we applied the prognostics algorithm RULCLIPPER introduced previously and the best subset was selected by minimising the scoring function.

3.8. Testing datasets

Given the training instances of a given dataset, the first task is to create a testing dataset in order to select the input features and the fusion rules. The training instances were truncated at a time instant randomly selected from a uniform distribution between 10% and 80% of the half-remaining life. This procedure allowed to obtain instances with small enough RULs to allow the occurrence of substantial degradation, and also large enough RULs to test the robustness of algorithms (Hu et al., 2012). The obtained testing datasets were used in RUL-CLIPPER with all subsets of features (511 subsets, 3168 fusion rules) and with two subsets of features (511×511 combinations for each fusion rule).

4. RESULTS AND DISCUSSIONS

Datasets $\#1, \#3$ are first considered (Section 4.2) followed by $\#2$ and $\#4$ (Section 4.3). An ensemble strategy is then proposed to improve results (Section 4.4). Results on the data challenge are finally presented (Section 4.5) with comparison (Section 3.3).
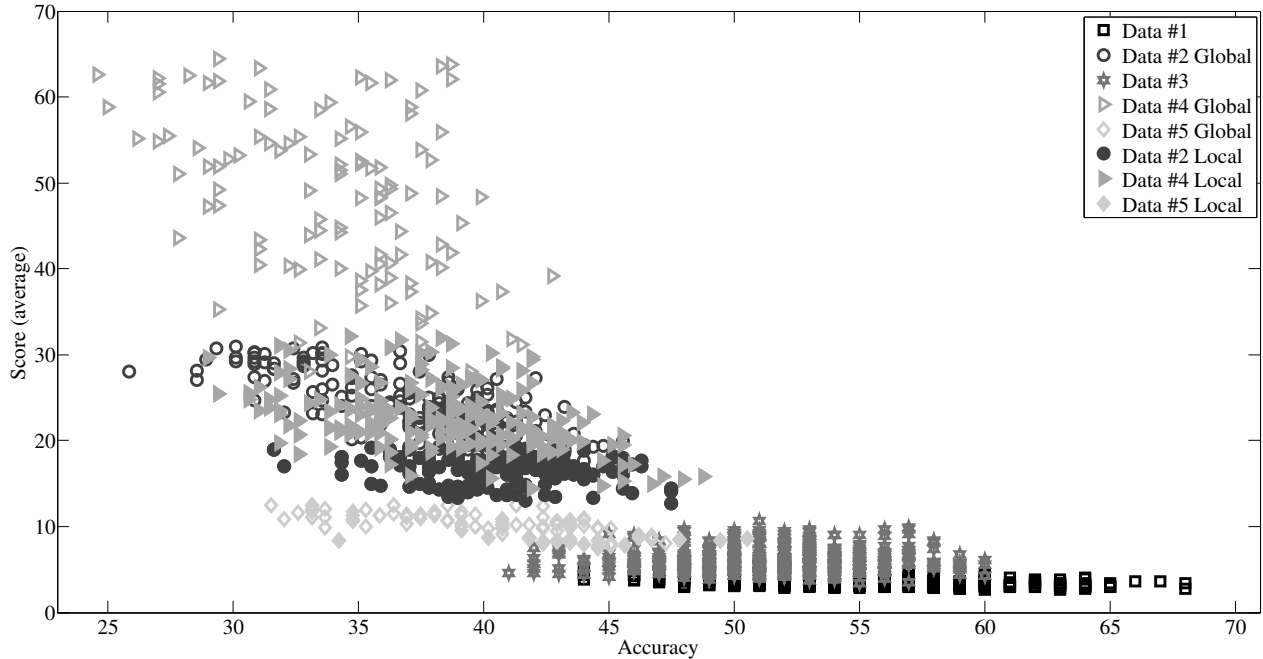
Figure 10. Performances in terms of accuracy (to be maximised) and score (to be minimised) for 511 combinations of features in each dataset where each point represents the best score (and its related accuracy) obtained for a particular subset of features.

4.1. Visualisation of results

The results are represented in the penalty – accuracy plane for all combinations of features. Two complementary views are proposed. The first one is depicted in Figure 10. Each type of marker represents a dataset and it is filled if the local approach was used for the estimation of the health indicator (otherwise the global approach was used). Each marker represents the performance for one subset and has two coordinates: The best score over all combination rules and its associated accuracy. Note that the latter can be smaller than the best accuracy. The utopic performance would correspond to a marker located at the bottom right-hand side ($S \to 0$, $A \to 100\%$).

This figure immediately demonstrates that the increasing complexity of datasets involves a degradation of the performances. Indeed, the points moves from the bottom right-hand side to the top left-hand side. RULCLIPPER performed well on datasets $\#1$, $\#3$, $\#5_T$ and less on $\#2$ and $\#4$. Therefore, it shows that the operating conditions have the greatest impact on the performances. The use of the local approach also greatly improved the performances (see filled markers). For example, results on dataset $\#4$ with the local approach are much better than the results on dataset $\#2$ with the global approach, while the latter is supposed easier than the former due to two fault modes.

Considering datasets $\#1$ and $\#3$, the cloud of markers does not show a large scattering on scores relatively to other datasets whereas the accuracy has much more deviation. For the other datasets, the scattering is more important which

means that the selection of the subset of sensors is more critical. These first observations argue in favor of using several performance measures to assess prognostics algorithms as suggested in (Saxena, Celaya, et al., 2008).

A complementary view is proposed in the next sections for a deeper analysis of results and illustrated in Figure 11. A point in the previous figure (Fig. 10) with coordinates ($P_1(S_1, A_1)$) is obtained by considering the accuracy (A_1) for the lowest (best) penalty score (S_1) given a subset of features. We propose to represent the imprecision concerning the performances by considering a second point $P_2^{'}(S_2, A_2)$ defined by the best accuracy A_2 obtained while keeping the score lower than the best score plus 25%, i.e. $S_2 \leq S_1 + 25\%S_1$ (see P_2 and $P_2^{'}$ in Figure 11) . These two points define a *rectangle* in the penalty – accuracy plane. These rectangles are then accumulated over all sensor subsets (right-hand side in Figure 11). In the ideal case, the algorithm would be insensitive to the choice of the subset of features (which are used to compute the health indicators) and therefore all subsets will lead to rectangles located at the same position. In a more realistic case, the accumulation may take the form of a unique cuboid when the sensitivity is limited (for example for $\#1$ and $\#3$) or multiple cuboids when the performances are dependent on the sensor subset (for example for $\#2$ and $\#4$).

In the following figures, the accumulation of rectangles are represented in gray scale. The whitest part corresponds to the area where most of rectangles are located and corresponding to the likeliest performances given several subsets of features.
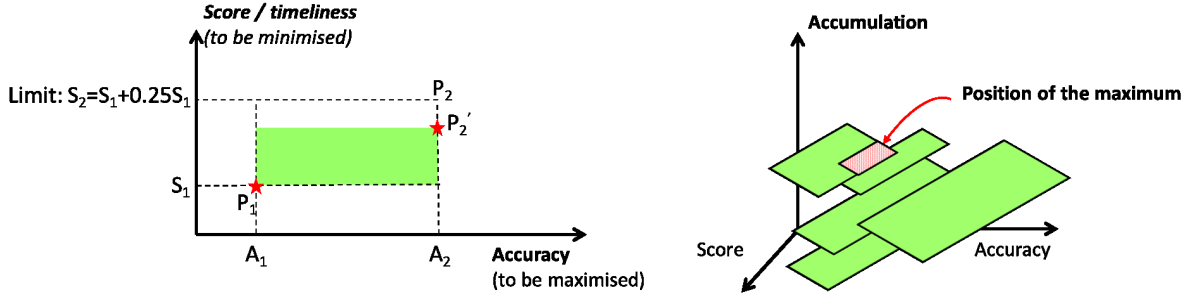
10

Figure 11. Evaluation of the sensitivity of sensor subsets on performance.

The scores have been divided by the number of testing instance for comparison purpose.

4.2. Performances on datasets #1 and #3

Figure 12 represents the accumulation of the rectangles for all combinations of features in the testing datasets #1 and #3. For dataset #1, the performance's centroid is located around $(60\%; 4.0)$ (or $(60\%; 400)$). One can draw any subset of features (among the 511 combinations considered) and can expect a score between $S = 310$ and $S = 440$ with an accuracy between $A = 56$ and $A = 64$. A few "optimal" subsets led to better performances (reported in Table 5 for the testing datasets). Due to the second fault mode, the scores are more scattered and a clear global decrease of the accuracy is observed (translation of the cluster of performances to the left hand-side). The level of the colorbar indicates that the choice of the features becomes more and more crucial as the difficulty of the dataset increases: It is simpler to find a subset of features for dataset #1 than for #3 leading to low penalty and high accuracy because the level is quite similar on a large area with less scattering (with a value around 8). It is more critical for dataset #3 since the cuboid is larger (in particular with respect to the timeliness) with a peak around 12 on a local area. A similar and magnified observation was obtained on the other datasets as shown in the next section.

Based on these results obtained on the testing datasets, the fusion rule and the subset of features were selected for final evaluation of the testing datasets by minimising the scoring function (as done for the PHM data challenge) and maximising the accuracy. The results obtained on the testing datasets #1 and #3 are summarised in Table 5 (note that the features indicated in the table have to be assembled with features 7, 8, 12, 16, 17, and 20). For each dataset, the combinations of features are given with respect to the two best scores ("Best S") and the two best accuracies ("Best A"). For example, the first line of Table 5 concerns dataset #1 for which the best score is $S = 261$ (with $A = 63\%$) when using features $9, 10, 14,$ 25 and 26, and the RUL fusion "$0.9mM(\hat{\text{RUL}}_{\mathcal{P}}^{th,11}) \oplus mw_7$" which corresponds to the combination of two elements: 1)
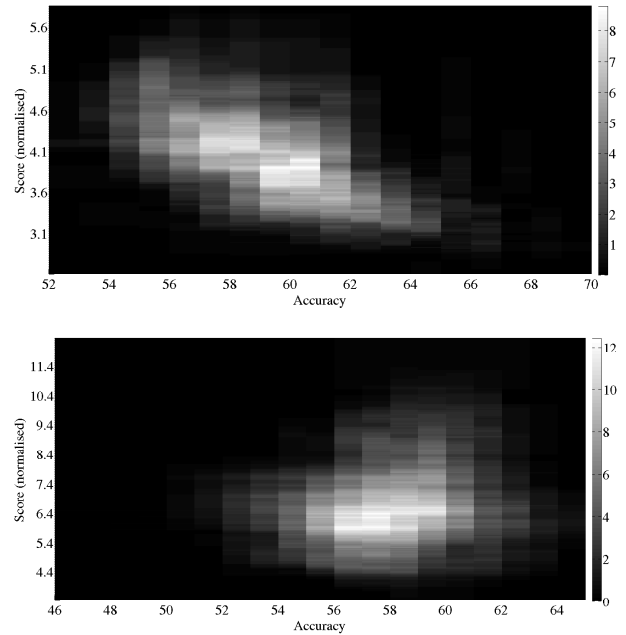

Figure 12. Performances for #1 (top) and #3 (bottom).

the output of the min/max operator (Eq. 13) with parameter $\alpha = 0.9$ applied on the 11 first RUL estimates and keeping only estimates with a similarity greater than 0.5, and 2) the weighted average (Eq. 16) of the $L = 7$ first RUL estimates (without outlier removal). The high value of α (0.9) implies more weight to the minimum (early) estimate meaning that the algorithm selects training instances which overestimate the RUL. An accuracy of 70% on #1 was obtained with the same subset of features while keeping a low score at $S = 301$. This accuracy obtained by the RULCLIPPER algorithm is significantly higher ($+16\%$) than the previous known results given by the EVIPRO-KNN algorithm (Ramasso et al., 2013) which yielded 53% or (Khelif et al., 2014) with 54%. Other metrics were computed (see Table 4, where metrics are defined in (Saxena, Celaya, et al., 2008)) for performance comparison with previous approaches: An exponential-based regression model with health indicator estimation proposed

in (Liu et al., 2013) reported a MAPE $= 9\%$ on #1 (compared to 20% for RULCLIPPER) and an Echo State Network with Kalman filter and submodels of instances presented in (Peng et al., 2012) with[1] MSE $= 3969$ (compared to 176 for RULCLIPPER).

The part entitled $(\#1, \#3)/S$ indicates the best scores for the same subset of features tested with the same fusion method on both datasets. Considering simultaneously #1 and #3 is equivalent to a situation where the engine is degrading while developing a fault. As the score is low and the accuracy high on both datasets using the same subset of features and the same method, it means that this parameterisation makes the prognostics robust to the introduction of a second fault mode.

4.3. Performances on datasets #2 and #4

These two datasets differ from the number of fault modes: one for #2 and two for #4. Moreover, compared to the two previous datasets, the difficulty increases by considering six operational conditions (Table 1) and 2.5 times more training and testing instances. Figure 13 represents the accumulation of the performance rectangles for all combinations of features in the testing datasets #2 and #4.
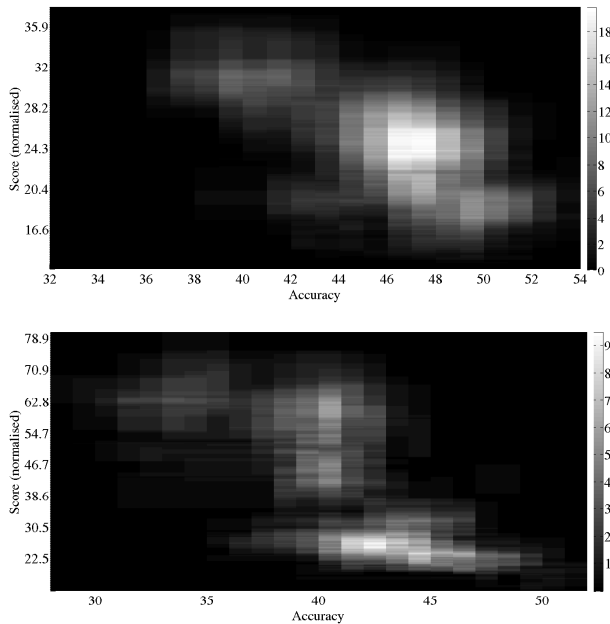

Figure 13. Performances for #2 (top) and #4 (bottom).

Compared to the two previous datasets, the whitest area (likeliest performances given several subset of features) is strongly shifted towards the upper left corner meaning that both the accuracy and score are degraded due to the presence of the operating conditions. For dataset #2, the performance's centroid

is located around $(47\%; 25)$ (or $(47\%; 6475)$ when not normalised). The level of the colorbar indicates that the choice of the features is still more critical compared to #3 with a level around 19 on a very local area. A few "optimal" subsets led to better performances (reported in Table 6 for the testing datasets).

For dataset #4 (the most difficult dataset), four performance's centroids can be found, in particular one is located around $(43\%; 27.5)$ (or $(43\%; 6820)$ when not normalised). For this dataset, one can not randomly draw a subset as it could be done with the other datasets. A few "optimal" subsets that led to the best performances are reported in Table 5.

4.4. RULCLIPPERs ensemble to manage sensors faults

As pointed out in (Simon, 2012), effective sensor selection tools are necessary to help end-users assess the health management consequences of adding or removing sensors and more generally to cope with sensor faults. One way to circumvent this issue is to consider *ensembles*: Several algorithms with different parameterisation are combined by expecting that the estimations will be improved and more reliable. Ensembles for prognostics have been considered in several papers, for instance in (P. Wang et al., 2012) applied to C-MAPSS datasets. In this paper, only two RULCLIPPERs were considered, each with one with a particular subset of features. For that, all couples of subsets of features were studied (about 130000 combinations) on each testing dataset. The best couples are given in Table 7.

Beyond the important improvement of scores and accuracies compared to the previous results (Table 5), it is interesting to notice that the best performances are not obtained by combining the two best parameterisations. Indeed, in most cases, the performances of RULCLIPPER with different parameterisations taken individually (with a given subset of features) are not the best ones, but their combination yielded to significative improvement of the performances compared to Table 5. For example, for dataset #1, combining RUL estimates provided by subset of features $\{10, 11, 14, 22\}$ (in addition to 7, 8, 12, 16, 17, 20) with $\{13, 18, 19, 22\}$ led to $S = 216$ and $P = 67\%$. It represents 27% of improvement on the score and $+4\%$ on accuracy compared to the best performances obtained in Table 5 with subset $\{13, 14, 18, 25\}$, and more when considering the performances of single subsets ($S_1 = 301$ and $P_1 = 64\%$, or $S_2 = 325$ with $P_2 = 62\%$). Similar observations can be made on the other datasets in particular on datasets #2 and #4 for which important improvements were obtained. A summary of results of RULCLIPPER on the turbofan datasets is given in Table 4 (metrics are defined in (Saxena, Celaya, et al., 2008)).

---

[1]Authors in (Peng et al., 2012) actually provided the best RMSE obtained equal to 63, so MSE was computed as $3969 = 63^2$.

RULCLIPPER performance						
Dataset	#1	#2	#3	#4	$\#5_T$	$\#5_V$
Score	216	2796	317	3132	752	11672
Accuracy (%)	67	46	59	45	n.a.	n.a.
FPR (%)	56	51	66	49	n.a.	n.a.
FNR (%)	44	49	34	51	n.a.	n.a.
MAPE (%)	20	32	23	34	n.a.	n.a.
MAE	10	17	12	18	n.a.	n.a.
MSE	176	524	256	592	n.a.	n.a.

Table 4. Summary of results of RULCLIPPER on all datasets. Metrics are defined in the glossary.

TYPE	DATA	FEATURES	FUSION	S	A
Best /S	#1	9, 10, 14, 25, 26	$0.9mM(\hat{\mathrm{RUL}}_{\mathcal{P}}^{th,11}) \oplus mw_7$	272	68
Best /A	#1	9, 10, 14, 25, 26	$0.9mM(\hat{\mathrm{RUL}}_{\mathcal{P}}^{th,11}) \oplus mw_{13}$	301	70
Best /S	#3	9, 13, 14, 22, 26	$0.9mM(\hat{\mathrm{RUL}}_{\mathcal{PK}}^{WL,11}) \oplus mw_3^{OR}$	353	57
Best /A	#3	9, 19, 25	$0.8mM(\hat{\mathrm{RUL}}_{\mathcal{PK}}^{WL,11})$	632	63
Best /A (2)	#3	18, 25, 26	$mw_{e,5}^{e,OR} \oplus mw_5$	580	63
Best /A (3)	#3	9, 10, 14, 18, 25	$0.8mM(\hat{\mathrm{RUL}}_{\mathcal{PK}}^{WL,11}) \oplus mw_3$	476	60
(#1, #3) /S	#1	9, 11, 14, 25, 26	$0.9mM(\hat{\mathrm{RUL}}_{\mathcal{P}}^{th,11}) \oplus mw_9^{e,OR}$	294	64
	#3	−	−	480	55
(#1, #3) /S(2)	#1	9, 10, 14, 25, 26	$0.9mM(\hat{\mathrm{RUL}}_{\mathcal{P}}^{th,11}) \oplus mw_{13}^{OR}$	299	63
	#3	−	−	480	56
(#1, #3) /S(2)	#1	9, 10, 14, 25, 26	$0.9mM(\hat{\mathrm{RUL}}_{\mathcal{P}}^{th,11}) \oplus mw_5^{OR}$	315	66
	#3	−	−	435	54

Table 5. Datasets #1, #3: Subset of features (in addition to 7, 8, 12, 16, 17, 20) leading to the best performances in terms of scores and accuracies for each dataset using a single RULCLIPPER. The rule "$0.9mM(\hat{\mathrm{RUL}}_{\mathcal{P}}^{th,11}) \oplus mw_7$" corresponds to the combination 1) the output of the min/max operator (Eq. 13) with $\alpha = 0.9$ applied on the 11 first RUL estimates with a similarity greater than 0.5, and 2) the weighted average (Eq. 16) of the $L = 7$ first RUL estimates (without outlier removal).

4.5. Results on the PHM'08 data challenge ($\#5_T$, $\#5_V$)

Based on the 218 training instances provided, RULCLIPPER was run on both testing datasets $\#5_T$ and $\#5_V$ called the PHM'08 data challenge, using the 511 combinations of features with the 3168 fusion rules. The results obtained were then sorted with respect to the scoring function. The first five best subsets of features were then selected: $\{9, 11, 26\}$, $\{9, 18, 22, 25\}$, $\{9\}$, $\{9, 10, 13, 25\}$, $\{9, 10, 18, 25, 26\}$ (in addition to $7, 8, 12, 16, 17, 20$ for each subset).

These combinations of features were considered and evaluated on the dataset $\#5_T$ (submitting estimations once a day on the NASA PCoE website). The best score was given by averaging three configurations of RULCLIPPERs, each with ensembles based on three subsets of features:

- RULCLIPPER 1 with inputs $\{9, 11, 26\}$, $\{9, 18, 22, 25\}$ and $\{9\}$;
- RULCLIPPER 2 with inputs $\{9, 11, 26\}$, $\{9, 18, 22, 25\}$ and $\{9, 10, 13, 25\}$;
- RULCLIPPER 3 with inputs $\{9, 11, 26\}$, $\{9, 18, 22, 25\}$ and $\{9, 10, 18, 25, 26\}$.

The RUL limit was set to 135 as described in Section 3.4 and the fusion rule was the same for all individual RULCLIPPER, namely $0.9mM(\hat{\mathrm{RUL}}_{\mathcal{PK}}^{11}) \oplus mw_{15}^{OR}$. The score obtained on dataset $\#5_T$ (on the NASA's website) was equal to 752, which is the 3rd score compared to published works. An alternative was considered by increasing the RUL limit from 135 to 175. The fusion rule was the same as previously and the score obtained was 934 which is quite low relatively to the high risk taken by setting the RUL limit to 175.

The average of the three configurations given above provided a set of RULs parameterised by both a RUL limit $(135, 175)$ and a fusion method. Three parameterisations were considered and combined:

- $\Omega_1 = (135, 0.8mM(\hat{\mathrm{RUL}}_{\mathcal{PK}}^{11}) \oplus mw_5^{OR})$,
- $\Omega_2 = (175, 0.9mM(\hat{\mathrm{RUL}}_{\mathcal{PK}}^{11}) \oplus mw_9^{OR})$, and
- $\Omega_3 = (175, 0.9mM(\hat{\mathrm{RUL}}_{\mathcal{PK}}^{11}) \oplus mw_{15})$.

The motivation of this configuration was to make long-term predictions possible while minimising the risk of making late predictions. The RULs obtained by Ω_2 and Ω_3 were averaged and the resulting combined by a weighted average with with Ω_1. The weights were set by a sigmoid (with shape parameter: 0.3 and position: 120) to increase the importance of RULCLIPPERs Ω_2 and Ω_3 when the estimation is greater than 120 while giving more importance to Ω_1 otherwise. The selected strategy for these datasets is depicted in Figure 14.

This methodology was then applied with the final testing

TYPE	DATA	FEATURES	FUSION	S	A
Best /S	#2	9, 13, 25	$mean(\hat{RUL}_{\mathcal{PK}}^{WL,11}) \oplus mw_3^{OR} \mid$ **L**	3296	47.5
	#2	–	$mean(\hat{RUL}_{\mathcal{PK}}^{WL,11}) \mid$ **L**	3354	47.5
	#2	–	$mw_3^{OR} \mid$ **L**	3443	41.3
Best /A	#2	9, 18, 22, 26	$0.8mM(\hat{RUL}_{\mathcal{PK}}^{WL,11}) \oplus mw_7^{OR} \mid$ **L**	4667	51.7
	#2	–	$0.8mM(\hat{RUL}_{\mathcal{PK}}^{WL,11}) \oplus mw_3^{e,OR} \mid$ **L**	4829	51.0
Best /S	#4	9, 10, 11, 22	$median(\hat{RUL}_{\mathcal{P}}^{OR,15}) \mid$ **L**	3576	41.9
Best /S (2)	#4	9, 10, 11, 22, 26	$0.9mM(\hat{RUL}_{\mathcal{PK}}^{WL,11}) \oplus mw_5 \mid$ **L**	3936	49.2
Best /A	#4	9, 11, 22	$0.9mM(\hat{RUL}_{\mathcal{P}}^{th,11}) \oplus mw_5^{OR} \mid$ **L**	4527	50.4
(#2, #4) /S	#2	9, 10, 13, 14, 26	$0.9mM(\hat{RUL}_{\mathcal{PK}}^{WL,11}) \oplus mw_3^{OR} \mid$ **L**	4096	42.1
	#4	–		5024	40.3
(#2, #4) /S(2)	#2	9, 13, 14	$0.8mM(\hat{RUL}_{\mathcal{PK}}^{WL,11}) \oplus mw_3^{OR} \mid$ **L**	3792	40.2
	#4	–		4465	41.9
(#2, #4) /S(3)	#2	9, 10, 18, 22, 26	$0.8mM(\hat{RUL}_{\mathcal{PK}}^{WL,11}) \oplus mw_3^{OR} \mid$ **L**	4138	50.2
	#4	–		4980	43.5
(#2, #4) /S(4)	#2	9, 10, 11, 25	$median(\hat{RUL}_{\mathcal{P}}^{OR,15}) \oplus mw_5^{OR} \mid$ **L**	4168	42.9
	#4	–		5116	46.0

Table 6. Datasets #2, #4: Subset of features (in addition to 7, 8, 12, 16, 17, 20) leading to the best performances in terms of scores and accuracies for each dataset using a single RULCLIPPER. **L** means that the local approach was used.
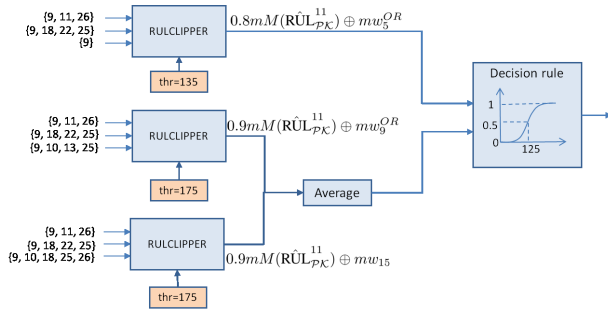

Figure 14. Methodology for $\#5_T$ and 5_V. The subsets of features (left hand side) have to be concatenated with $7, 8, 12, 16, 17, 20$.

dataset ($\#5_V$) yielding 11672. The comparison with approaches can be quantified on Figure 15. The generalisation of RULCLIPPER parameterised as proposed in this paper is lower than the first five algorithms (see square markers on the left-hand side). Indeed, some of these algorithms provided higher scores on $\#5_T$ but lower on the final dataset $\#5_V$. One explanation accounting for the lack of generalisation capability compared to the first five algorithms may hold in the "rules" integrated in RULCLIPPER (section 3.4). These rules have been tuned according to observations on the five other datasets but may be not relevant for dataset $\#5_V$ if the statistics governing the generation of instances have been modified (Saxena, Goebel, et al., 2008). In order to show the applicability of RULCLIPPER algorithm with as less parameterisation as possible, the author intentionally kept the same settings for all datasets without distinction in particular concerning the number of fault modes or thresholds on RUL limits. The authors also remarked on the previous datasets (#1 to #4) that a few instances can disturb the algorithm (in particular to test the robustness), generating very late or very early predictions, degrading drastically the scores.

However, the generalisation is better than the 23 remaining algorithms, for which lower scores on $\#5_T$ have been obtained with higher ones on $\#5_V$ (see square markers on the right-hand side). RULCLIPPER provided a relatively low score on both datasets using the same parameters (816 on $\#5_T$ and 11672 on $\#5_V$). Scores obtained on the turbofan datasets are also good compared to the state-of-the-art (Tables 2, 3 and 4).

5. CONCLUSION

The RULCLIPPER algorithm is proposed to estimate the remaining lifetime of systems in which noisy health indicators are assumed imprecise. It is made of elements inspired from both the computer vision and computational geometry communities and relies on the adaptation of case-based reasoning to manage the imprecise training and testing instances. The combination of these elements makes it an original and efficient approach for RUL estimation.

RULCLIPPER was validated with the six datasets coming from the turbofan engine simulator (C-MAPSS), including the so-called turbofan datasets (four datasets) and the data challenge (two datasets), and compared to past work. These datasets are considered as complex due to the presence of fault modes and operating conditions. In addition to RULCLIPPER, a method was proposed to estimate the health indicator (in presence of faults and operating conditions) and the problem of the selection of the most relevant sensors was also tackled. Information fusion rules were finally studied to combine RUL estimates as well as ensemble of RULCLIPPERs. The review of past work (a detailed survey is in under preparation), the presentation of the datasets, as well as the
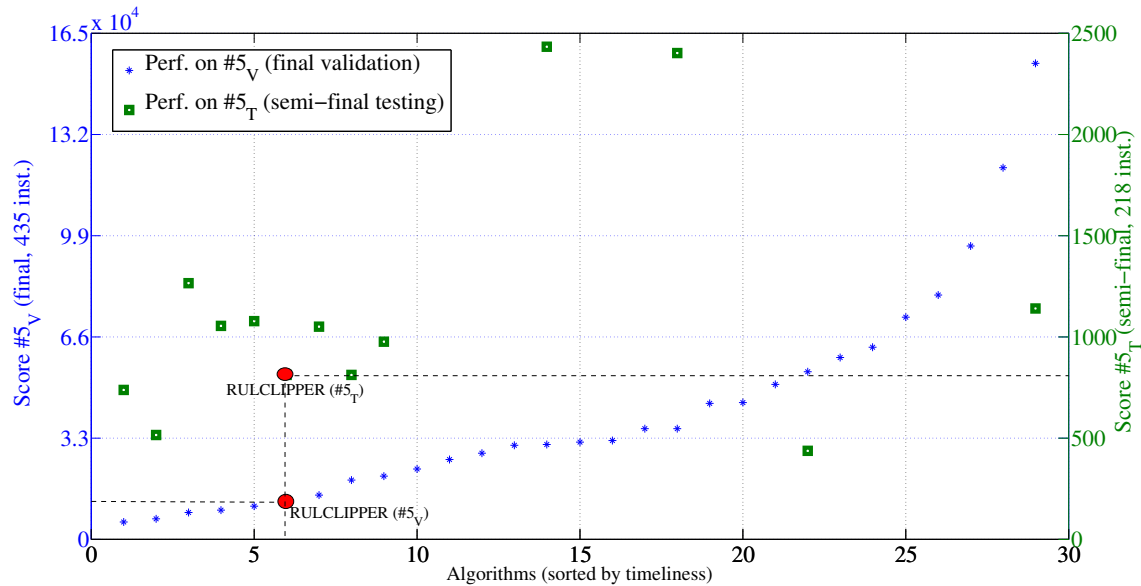
Figure 15. Comparison of RULCLIPPER with other state-of-the-art approaches. Some scores on the testing dataset $\#5_T$ are missing. Scores have to be minimised.

results on sensor selection, health indicator estimation, information fusion rules and RULCLIPPERs ensemble are expected to give a hand to other researchers interested in testing their algorithms on these datasets.

RULCLIPPER illustrates that computational geometry seems promising for PHM in presence of noisy HIs. While some similarity-based matching algorithms may suffer from computational complexity in particular to sort training instances, RULCLIPPER is characterised by fast operations: Intersection of IHI with length close to 220 units (among the longest ones) took only 3.8 *milli*seconds (less than 2 minutes for dataset #1). Computational geometry has become an active field in particular to improve memory and time requirements, with applications in multimedia for which CUDA implementations on processor arrays on graphic cards were proposed. With such implementations, real-time and anytime prognostics can be performed. The extension to multiple health indicators is under study by using polytopes.

ACKNOWLEDGMENT

NOMENCLATURE

A	Accuracy
CBR	Case-based reasoning
C-MAPSS	Commercial Modular Aero-Propulsion System Simulation
DW/OW/AW	Diversity / Optimization / Accuracy-based weighting
EXP	Least-square exponential fitting
FPR/FNR	False positive/negative rate
GPM	General Path Model
(I)HI	(Imprecise) Health indicator
MAPE	Mean absolute percentage error
MLP	MultiLayer Perceptron
	heaLth Indicator modeled by Planar Polygons and similarity-basEd Reasoning
MSE/MAE	Mean squared/absolute error
OR	Outlier removal
PHM	Prognostics and Health Management
QUAD	Quadratic fitting
REG	Linear regression
RVM/SVM	Relevance/Support vector machine
RNN	Recurrent neural network
RUL	Remaining Useful Life
RULCLIPPER	RUL estimation based on impreCise
S	Score (timeliness)
SBI	Similarity-based interpolation
SBL	Sparse Bayesian Learning
WL	Whisker length

15

Data	Type	Features						Fusion	S	P
		Subset 1	S_1	P_1	Subset 2	S_2	P_2			
#1	Best /S	10, 11, 14, 22	301	64 **L**	13, 18, 19, 22	325	62	$0.9mM(\hat{RUL}_{P}^{th,11}) \oplus mw_5^{OR}$	**216**	67
	Best /A	10, 11, 14, 22	301	64	13, 18, 19, 22	325	62	$0.9mM(\hat{RUL}_{P}^{th,11}) \oplus mw_7^{OR}$	224	**70**
#2	Best /S	9, 13, 14, 26	3782	42.9 **L**	9, 10, 13, 26	3538	49.0 **L**	$0.7mM(\hat{RUL}_{PK}^{WL,15}) \oplus mw_5^{OR}$	**2796**	45.6
	Best /S (2)	9, 18, 22	4331	51.4 **L**	9, 10, 13, 25	3451	50.1 **L**	$mea(\hat{RUL}_{PK}^{WL,15})$	2905	51.0
	Best /A	9, 10, 13, 22, 25	3539	50.58 **L**	9, 10, 13, 26	3538	49.0 **L**	$0.8mM(\hat{RUL}_{PK}^{WL,15}) \oplus mw_5^{OR}$	3498	**52.5**
	Best /A (2)	9, 10, 18, 22	4122	49.8 **L**	9, 10, 13, 25	3451	50.1 **L**	$mea(\hat{RUL}_{PK}^{WL,15})$	2945	51.7
#3	Best /S	13, 19, 25, 26	525	61	9, 13, 14, 22, 26	353	58	$0.9mM(\hat{RUL}_{P}^{th,15})$	**317**	59
	Best /A	14, 18, 19, 26	499	61	9, 10, 13, 14, 26	399	58	$0.9mM(\hat{RUL}_{P}^{WL,15}) \oplus mw_{11}^{OR}$	332	**63**
#4	Best /S	9, 10, 11, 25, 26	5180	48.8 **L**	9, 13, 22, 25	5748	46.0 **L**	$0.9mM(\hat{RUL}_{P}^{th,11}) \oplus mw_{11}$	**3132**	44.8
	Best /S (2)	9, 10, 11, 19, 22	4847	45.6 **L**	9, 11, 22	3655	50.4 **L**	$0.9mM(\hat{RUL}_{P}^{WL,15}) \oplus mw_5^{OR}$	3275	50.9
	Best /A	9, 19	6618	43.6 **L**	9, 11, 22, 26	3843	50.8 **L**	$0.8mM(\hat{RUL}_{PK}^{WL,15}) \oplus mw_3^{OR}$	4581	**54.1**
	Best /A (2)	9, 19	6618	43.6 **L**	9, 11, 22	3655	50.4 **L**	$0.8mM(\hat{RUL}_{PK}^{WL,15}) \oplus mw_3^{OR}$	4194	53.2

Table 7. RULCLIPPERs ensemble: Combination of subsets of features (in addition to 7, 8, 12, 16, 17, 20) leading to the best performances.

REFERENCES

Al-Salah, T., Zein-Sabatto, S., & Bodruzzaman, M. (2012). Decision fusion software system for turbine engine fault diagnostics. In *Proc. of IEEE southeastcon* (p. 1-6).

Beer, M., Ferson, S., & Kreinovich, V. (2013). Imprecise probabilities in engineering analyses. *Mechanical Systems and Signal Processing*, *37*(1-2), 4-29.

Bentley, J., & Ottmann, T. (1979). Algorithms for reporting and counting geometric intersections. *IEEE Trans. Comput.*, *C28*, 643-647.

Bonissone, P., Varma, A., & Aggour, K. (2005). A fuzzy instance-based model for predicting expected life: A locomotive application. In *IEEE int. conf. on computational intelligence for measurement systems and applications* (p. 20-25).

Celaya, J., Kulkarni, C., Biswas, G., & Goebel, K. (2012). Towards a model based prognostics methodology for electrolytic capacitors a case study based on electrical overstress accelerated aging. *Int. Journal of Prognostics and Health Management*, *3*(2-004), 1-19.

Chazelle, B., & Edelsbrunner, H. (1992). An optimal algorithm for intersecting line segments in the plane. *J. Assoc. Comput. Mach*, *39*, 1-54.

Chen, C., Zhang, B., Vachtsevanos, G., & Orchard, M. (2011). Machine condition prediction based on adaptive neuro-fuzzy and high-order particle filtering. *IEEE Trans. on Industrial Electronics*, *58*(9), 4353-4364.

Coble, J. (2010). *Merging data sources to predict remaining useful life - an automated method to identify prognostic parameters* (Unpublished doctoral dissertation). University of Tennessee, Knoxville.

Coble, J., & Hines, J. W. (2011). Applying the general path model to estimation of remaining useful life. *Int. Journal on Prognostics and Health Management*, *2*, 326-337.

Daigle, M., & Goebel, K. (2011). A model-based prognostics approach applied to pneumatic valves. *Int. Journal of Prognostics and Health Management*, *2*(2-008), 1-16.

Filippov, A. (1950). An elementary proof of Jordan's theorem. *Uspekhi Mat. Nauk*, *5*, 173-176.

Gelman, I., Patel, T., Murray, B., & Thomson, A. (2013). Rolling bearing diagnosis based on the higher order spectra. *Int. Journal of Prognostics and Health Management*, *4*(2-022), 1-9.

Goharrizi, A. Y., & Sepehri, N. (2010). A wavelet-based approach to internal seal damage diagnosis in hydraulic actuators. *IEEE Trans. on Industrial Electronics*, *57*(5), 1755-1763.

Gouriveau, R., Ramasso, E., & Zerhouni, N. (2013). Strategies to face imbalanced and unlabelled data in PHM applications. *Chemical Engineering Trans.*, *33*, 115-120.

Gouriveau, R., & Zerhouni, N. (2012). Connexionist-systems-based long term prediction approaches for prognostics. *IEEE Trans. on Reliability*, *61*, 909-920.

Greiner, G., & Hormann, K. (1998). Efficient clipping of arbitrary polygons. *ACM Trans. on Graphics*, *17*, 71-83.

Gucik-Derigny, D., Outbib, R., & Ouladsine, M. (2011). Prognosis applied to an electromechanical system, a nonlinear approach based on sliding mode observer. In *Annual conf. on european safety and reliability association.*

Gustafson, E., & Kessel, W. (1978). Fuzzy clustering with a fuzzy covariance matrix. In *IEEE conf. on decision and control.*

He, D., & Bechhoefer, E. (2008). Development and validation of bearing diagnostic and prognostic tools using HUMS condition indicators. In *IEEE aerospace conf.*

He, D., Bechhoefer, E., & Saxena, A. (2013). Editorial: Special issue on wind turbine phm. *Int. Journal of Prognostics and Health Management*, *4*(4-031), 2.

Heimes, F. (2008). Recurrent neural networks for remaining useful life estimation. In *IEEE int. conf. on prognostics and health management.*

Hu, C., Youn, B., Wang, P., & Yoon, J. (2012). Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. *Reliability Engineering and System Safety*, *103*, 120 - 135.

Javed, K., Gouriveau, R., & Zerhouni, N. (2013). Novel failure prognostics approach with dynamic thresholds for machine degradation. In *IEEE industrial electronics conf.*

Khelif, R., Malinowski, S., Morello, B., & Zerhouni, N. (2014). Rul prediction based on a new similarity-instance based approach. In *IEEE ISIE.* Istambul, Turkey.

Klir, G., & Wierman, M. (1999). Uncertainty-based information. elements of generalized information theory. In (chap. Studies in fuzzyness and soft computing). Physica-Verlag.

Kuncheva, L. I. (2004). *Combining pattern classifiers: Methods and algorithms*. Wiley-Interscience.

Lapira, E., Siegel, D., Zhao, W., Brisset, D., Su, J., Wang, C., . . . Lee, J. (2011). A systematic framework for wind turbine health assessment under dynamic operating conditions. In *24th int. congress on condition monitoring and diagnostics engineering management.*

Liu, K., Gebraeel, N. Z., & Shi, J. (2013). A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis. *IEEE Trans. on Automation Science and Engineering.*

Longley, P., de Smith, M., & Goodchild, M. (2007). *Geospatial analysis : A comprehensive guide to principles, techniques and software tools*. Matador, Leicester.

Margalit, A., & Knott, G. (1989). An algorithm for computing the union, intersection or difference of two polygons. *Computers & Graphics*, *13*, 167-183.

Mulligan, K., Yang, C., Quaegebeur, N., & Masson, P. (2013). A data-driven method for predicting structural degradation using a piezoceramic array. *Int. Journal of Prognostics and Health Management*, *4*(2-037), 1-14.

Orchard, M., Cerda, M., Olivares, B., & Silva, J. (2012). Sequential monte carlo methods for discharge time prognosis in lithium-ion batteries. *Int. Journal of Prognostics and Health Management*, *3*(2-010), 1-12.

Orchard, M., Kacprzynski, G., Goebel, K., Saha, B., & Vachtsevanos, G. (2008). Advances in uncertainty representation and management for particle filtering applied to prognostics. In *Int. conf. on prognostics and health management.*

Peel, L. (2008). Data driven prognostics using a Kalman filter ensemble of neural network models. In *Int. conf. on prognostics and health management.*

Peng, T., He, J., Liu, Y., Saxena, A., Celaya, J., & Goebel, K. (2012). Integrated fatigue damage diagnosis and prognosis under uncertainties. In *Annual conf. of prognostics and health management.*

Powers, D. (2011). Evaluation: From precision, recall and F-factor to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, *2*, 37-63.

Ramasso, E. (2014). Investigating computational geometry for failure prognostics in presence of imprecise health indicator: results and comparisons on C-MAPSS datasets. In *European conference of the prognostics and health management society* (Vol. 5, p. 1-13).

Ramasso, E., & Denoeux, T. (2013). Making use of partial knowledge about hidden states in hidden Markov models: an approach based on belief functions. *IEEE Trans. on Fuzzy Systems*, *22*(2), 395-405.

Ramasso, E., & Gouriveau, R. (2013). RUL estimation by classification of predictions: an approach based on a neuro-fuzzy system and theory of belief functions. *IEEE Trans. on Reliability*, *Accepted*.

Ramasso, E., Rombaut, M., & Zerhouni, N. (2013). Joint prediction of observations and states in time-series based on belief functions. *IEEE Trans. on Systems, Man and Cybernetics - Part B: Cybernetics*, *43*, 37-50.

Riad, A., Elminir, H., & Elattar, H. (2010). Evaluation of neural networks in the subject of prognostics as compared to linear regression model. *Int. Journal of Engineering & Technology*, *10*, 52-58.

Richter, H. (2012). Engine models and simulation tools. In *Advanced control of turbofan engines* (p. 19-33). Springer New York.

Rigaux, P., Scholl, M., & Voisard, A. (2002). *Spatial databases with application to gis* (E. Science, Ed.). Kauffman Publishers.

Rosen, K. (2004). *Handbook of discrete and computational geometry, second edition* (J. E. Goodman & J. O'Rourke, Eds.). Chapman and Hall/CRC.

Saha, B., Goebel, K., & Christophersen, J. (2008). Comparison of prognostic algorithms for estimating remaining useful life of batteries. *Trans. of the Royal UK Institute on Measurement & Control, special issue on Intelligent Fault Diagnosis & Prognosis for Engineering Systems*, 293-308.

Saha, B., Goebel, K., Poll, S., & Christophersen, J. (2009). Prognostics methods for battery health monitoring using a Bayesian framework. *IEEE Trans. on Instrumentation and Measurement*, *58*, 291-296.

Sankararaman, S., Ling, Y., Shantz, C., & Mahadevan, S. (2009). Uncertainty quantification in fatigue damage prognosis. In *Annual conf. of the prognostics and health management society*.

Sarkar, S., Jin, X., & Ray, A. (2011). Data-driven fault detection in aircraft engines with noisy sensor measurements. *Journal of Engineering for Gas Turbines and Power*, *133*, 081602.

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008). Metrics for evaluating performance of prognostic techniques. In *Int. conf. on prognostics and health management* (p. 1-17).

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *Int. conf. on prognostics and health management* (p. 1-9). Denver, CO, USA.

Saxena, A., Wu, B., & Vachtsevanos, G. (2005). Integrated diagnosis and prognosis architecture for fleet vehicles using dynamic case-based reasoning. In *Autotestcon* (p. 96-102).

Serir, L., Ramasso, E., & Zerhouni, N. (2013). E2GKpro: An evidential evolving multi-modeling approach for system behavior prediction with applications. *Mechanical Systems and Signal Processing*, *37*(1-2), 213-228. doi: 10.1016/j.ymssp.2012.06.023

Simon, D. (2012). *Challenges in aircraft engine gas path health management.* (Tutorial on Aircraft Engine Control and Gas Path Health Management Presented at 2012 Turbo Expo)

Vachtsevanos, G. (2006). Intelligent fault diagnosis and prognosis for engineering systems. Wiley, Hoboken, NJ.

Vatti, B. R. (1992). A generic solution to polygon clipping. *Communications of the ACM*, *35*, 56-63.

Wang, P., Youn, B., & Hu, C. (2012). A generic probabilistic framework for structural health prognostics and uncertainty management. *Mechanical Systems and Signal Processing*, *28*, 622 - 637.

Wang, T. (2010). *Trajectory similarity based prediction for remaining useful life estimation* (Unpublished doctoral dissertation). University of Cincinnati.

Wang, T., Yu, J., Siegel, D., & Lee, J. (2008). A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *Ieee int. conf. on prognostics and health management* (p. 1-6).

Zein-Sabatto, S., Bodruzzaman, J., & Mikhail, M. (2013). Statistical approach to online prognostics of turbine engine components. In *Proc. of IEEE southeastcon* (p. 1-6).

Zhang, X., & Pisu, P. (2014). Prognostic-oriented fuel cell catalyst aging modeling and its application to health-monitoring and prognostics of a PEM fuel cell. *Int. Journal of Prognostics and Health Management*, *5*(1-003), 1-16.

BIOGRAPHY

Dr. Emmanuel Ramasso received the B.Sc. and M.Sc. degrees in Automation Science and Engineering from the University of Savoie in 2004, and earned his Ph.D. from the University of Grenoble in 2007. He pursued with a postdoc at the Commissariat à l'Energie Atomique et aux Energies Alternatives in 2008. Since 2008, he has been working as an associate professor at the National School of Engineering in Mechanics and Microtechnics (ENSMM) at Besançon (France). His research is carried out at FEMTO-ST institute and focused on pattern recognition under uncertainties, Prognostics and Health Management (PHM) and Structural Health Management (SHM).